ISSN 1561 – 5359. Штучний інтелект, 2015, № 1-2

УДК 004.9 : 519.8

В.В. Любченко
Одеський національний політехнічний університет, Україна
Україна, 65044, м. Одеса, пр. Шевченка, 1

СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ЗАДАЧІ НАСТУПНОГО РЕЛІЗУ

V. Liubchenko
Odessa National Polytechnic University, Ukraine
Ukraine 65044, Odessa, Shevchenko av., 1

DECISION SUPPORT SYSTEM FOR NEXT RELEASE PROBLEM

В.В. Любченко
Одесский национальный политехнический университет, Украина
Украина, 65044, г. Одесса, пр. Шевченко, 1

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ ЗАДАЧИ СЛЕДУЮЩЕГО РЕЛИЗА

Розглядається задача наступного релізу у вигляді задачі оптимізації з двома цільовими функціями. Визначено два види залежностей між вимогами та спосіб їх урахування при вирішенні задачі наступного релізу. Для побудови Парето-фронту використовуються евристичні методи, розглянуті обмеження їх застосування. Розроблена структуру системи підтримки прийняття рішень для задачі наступного релізу.
Ключові слова: система підтримки прийняття рішень, задача наступного релізу, двокритеріальна оптимізація.

The next release problem is examined as the bi-objective optimization problem. Two types of dependencies between requirements are identified and their influence on the next release problem formulation is analyzed. Heuristic methods are used for construction of the Pareto-front, their constraints are described. The structure of decision support systems for the next release problem is proposed.
Key words: decision support system, next release problem, bi-objective optimization.

Рассматривается задача следующего релиза в виде задачи оптимизации с двумя целевыми функциями. Определены два вида зависимостей между требованиями и способ их учета при решении задачи следующего релиза. Для построения Парето-фронта используются эвристические методы, рассмотрены ограничения их применения. Разработана структура системы поддержки принятия решений для задачи следующего релиза.
Ключевые слова: система поддержки принятия решений, задача следующего релиза, двухкритериальная оптимизация.

Критичні помилки в процесі розробки програмного забезпечення, такі як надмірно велика множина вимог для реалізації протягом релізу, часто зумовлені перебільшенням довіри до інтуїції особи, що приймає рішення (ОПР). У таких випадках доцільним є використання систем підтримки прийняття рішень (СППР), метою яких є допомога ОПР в складних умовах для повного і об'єктивного аналізу предметної діяльності. Особливо важливим є застосування СППР на ранніх етапах життєвого циклу програмного забезпечення (ПЗ), коли приймаються рішення високого рівня, які впливають на подальший процес розробки ПЗ.

Одним з подібних рішень є вирішення задачі наступного релізу (ЗНР), яка виникає у випадку ітеративної розробки ПЗ. Ітеративна розробка – це методологія, в якій послідовно повторюються процеси з розробки програмного продукту і безперервного аналізу отриманих результатів з коригуванням наступних етапів

 ©В.В. Любченко

роботи. На початку розробки програмного продукту складається специфікація вимог до нього, до якої протягом розробки можуть вноситися зміни та доповнення. Задача ОПР полягає у визначенні на початку кожної ітерації підмножини вимог, які мають розроблятися на наступній ітерації для того, щоб задовольнити найбільше потреб споживачів за умови мінімізації витрачених коштів розробника ПЗ [1]. Для забезпечення розуміння ситуації для ОПР може бути застосована пошукова оптимізація [2]. Оскільки ЗНР є різновидом аналізу зисків і витрат, її рішення має бути оптимальним за Парето [3].

Сформулюємо ЗНР математично. Хай є множина споживачів ПЗ $C = \{c_1, ..., c_m\}$, кожен з яких характеризується важливістю для компанії, що відображається множиною вагових коефіцієнтів $Weight = \{w_1, ..., w_m\}$, де $w \in [0,1]$ та $\sum_{j=1}^{m} w_j = 1$. Припустимо також, що є множина вимог $R = \{r_1, ..., r_n\}$, які треба задовольнити в наступних релізах існуючого ПЗ. Не будемо розрізняти функціональні та нефункціональні вимоги, оскільки всі вони зумовлюють корисність ПЗ для споживачів [4]. Програмна реалізація кожної вимоги потребує певних ресурсів, які можуть бути оцінені в термінах вартості $Cost = \{cost_1, ..., cost_n\}$.

Реалізація вимог забезпечує зиски для компанії. Рівень задоволення окремих споживачів залежить від того, яка підмножина вимог буде реалізована в наступному релізі програмного продукту. Важливість вимог не однакова для різних споживачів. Кожен споживач c_j ($1<j<m$) встановлює для вимоги r_i ($1<i<n$) значення $value(r_i, c_j)$, при цьому $value(r_i, c_j)>0$, коли споживач j зацікавлений у вимозі i, та 0 – в іншому випадку.

Вектор рішення $\vec{x} = (x_1, ..., x_n)$, в якому $x_i \in \{0,1\}, i = \overline{1,n}$, визначає, які з вимог будуть обрані в розробку на наступній ітерації. Елемент x_i дорівнює 1, якщо вимога i включена до розробки на наступній ітерації, і дорівнює 0 в іншому випадку.

У більшості випадків між вимогами до ПЗ існують певні залежності. Введемо у розгляд два типи відношень, які моделюють залежності між вимогами:

– відношення передування $Prec(r_i,r_j)$, яке означає, що вимога r_i має бути реалізована не пізніше вимоги r_j;

– відношення одночасності $Coup(r_i,r_j)$, яке означає, що вимога r_i має бути реалізована на тій самій ітерації, що і вимога r_j.

Врахування існування залежностей між вимогами призводить до необхідності внесення певних змін в оцінки, що характеризують вимоги. Через необхідність реалізації вимог в певному порядку, змінюється оцінка вартості їх виконання на наступній k-й ітерації:

$$cost_j^k = cost_j + \sum_{\forall i : (Prec(r_i, r_j) \vee Coup(r_i, r_j)) \wedge (x_i = 0)} cost_i , \tag{1}$$

тобто за умови існування вимог, які мали бути реалізовані раніше або одночасно, для аналізованої вимоги вартість реалізації збільшується.

Також необхідно врахувати, що реалізація вимоги без урахування існуючих взаємозв'язків не принесе користі для споживачів, тому зиск від реалізації вимог на k-й ітерації визначається як:

$$value^k(r_j, c_l) = \begin{cases} value(r_j, c_l) \text{ якщо } \{r_i : (Prec(r_i, r_j) \vee Coup(r_i, r_j)) \wedge (x_i = 0)\} = \varnothing \\ 0, \text{ в іншому випадку} \end{cases} , \tag{2}$$

Очевидно, що при рішенні ЗНР слід взяти до уваги дві цільові функції. Перша з них відповідає загальному зиску для компанії, якій очікується отримати як результат k-ї ітерації:

$$\mathrm{f}_1(\vec{x}) = \sum_{i=1}^{n} x_i \sum_{j=1}^{m} w_j \cdot value^k(r_i, c_j). \tag{3}$$

Задача полягає у виборі підмножини вимог, яка призведе до максимізації зиску для компанії.

Друга цільова функція відповідає загальним коштам, потрібним для задоволення вимог на k-й ітерації:

$$\mathrm{f}_2(\vec{x}) = \sum_{i=1}^{n} cost_i^k \cdot x_i . \tag{4}$$

Задача полягає у виборі підмножини вимог, яка призведе до мінімізації коштів компанії.

Відомо [3], що Парето-фронт визначається як $P(Y) = \{y \in Y : \{y' \in Y : y' \succ y\} = \varnothing\}$, де відношення домінування по Парето $y' \succ y$ означає, що $y \le y', y \ne y'$. Тому задача мінімізації, за другим критерієм, має бути перетворена на задачу максимізації: помножимо цільову функцію на -1. Тепер ЗНР для k-ї ітерації можна записати в такому виді:

$$\begin{array}{l} \text{Maximize } \mathrm{f}_1(\vec{x}) = \sum_{i=1}^{n} x_i \sum_{j=1}^{m} w_j \cdot value^k(r_i, c_j) \\ \text{Maximize } \mathrm{f}_2(\vec{x}) = -\sum_{i=1}^{n} cost_i^k \cdot x_i \end{array} . \tag{5}$$

Таким чином, ЗНР є задачею оптимізації з двома цільовими функціями на комбінаторній множині комбінацій вимог.

Сформульована двокритеріальна задача (5) є NP-трудною, тому не може бути вирішена за допомогою точних методів оптимізації через великий розмір комбінаторної множини комбінацій вимог. Тому для апроксимації множини Парето-оптимальних рішень двокритеріальної задачі зазвичай застосовують евристичні пошукові техніки. У [2] показано, що застосування методу NSGA-II призводить до отримання найбільшої кількості оптимальних рішень, метод MOCell забезпечує ОПР найрізноманітнішими рішеннями, а метод PAES працює найшвидше, проте забезпечує найменшу кількість оптимальних рішень.

Обмеження для застосування цього підходу пов’язані з розміром ЗНР. Задача може бути занадто мала для застосування евристичних методів. Емпіричні дослідження показали, що для пошуку рішення має бути визначено і оцінено більше ніж 20 вимог, натомість немає занадто малої кількості споживачів.

Але також задача може бути занадто велика для рішення через обмежені обчислювальні можливості СППР. У [5] розглянуто три варіанти алгоритму побудови Парето-фронту для застосування їх з метою вирішення ЗНР на основі моделі двокритеріальної оптимізації, та показано, що алгоритм повного перебору з фільтруванням є кращим алгоритмом за критеріями часу розрахунків і витрат пам'яті. Цей алгоритм відрізняться тим, що на етапі підготовки підмножин вимог для застосування до них евристик, підмножини-кандидати проходять фільтрацію на

основі оцінок їх зиску. Для застосування залишаються лише ті з них, які найбільш близькі до Парето-фронту. Додатково для зменшення розмірності ЗНР можна врахувати статистичні дані про витрати ресурсів протягом попередніх ітерацій.

Тепер можемо визначити структуру СППР для ЗНР (рис. 1).



Рис.1. Структура СППР для ЗНР

Структуру СППР складають 4 підсистеми. Редактор вимог дозволяє відповідальній особі (власнику продукту, бізнес аналітику, тощо) додавати, змінювати та видаляти данні про вимоги до програмного продукту. Підсистема оцінювання вимог дозволяє підтримувати в актуальному стані оцінки важливості окремих споживачів, а також оцінки вартості ресурсів та зиску від реалізації окремих вимог. Як вимоги, так і оцінки можуть змінюватися протягом процесу розробки. Усі ці дані зберігаються в базі даних проекту розробки ПЗ та використовуються підсистемою визначення Парето-фронту для задачі (5). ОПР обирає рішення з запропонованої множини відповідно до власних преференцій.

При застосуванні СППР для ЗНР треба розуміти, що рішення, які вона генерує, дуже залежать від очікувань та припущень, що закладені до цільових функцій. Незважаючи на це, є багато прикладів того, що застосування методів пошукової оптимізації дає можливість системно проаналізувати комбінаторну множину комбінацій вимог та отримати, як рекомендовані, не лише очевидні, а й також неочікувані рішення. Застосування цих методів є ефективним у поєднанні з застосуванням знань та інтуїції ОПР.

Литература

1. Durillo, J. J. A study of the bi-objective next release problem / J. J. Durillo, Y. Zhang, E. Alba, M. Harman, A. J. Nebro // Empirical Software Engineering. – 2011. – Vol. 16 (1). – P. 29 – 60.
2. Harman, M. Search based software engineering / M. Harman, B. F. Jones // Information and Software Technology. – 2001. – № 43 (14). – P. 833 – 839.
3. Ногин, В. Д. Принятие решений в многокритериальной среде: количественный подход / В. Д. Ногин. – М. : ФИЗМАТЛИТ, 2002. – 176 с.
4. Денисюк, А. В. Актуальные проблемы качества программного обеспечения / А. В. Денисюк, В. В. Любченко // Электротехнические и компьютерные системы. – 2013. – № 9 (85). – С. 142 – 148.
5. Любченко В. В. Алгоритм знаходження Парето-оптимального рішення задачі наступного релізу / В. В. Любченко, Є. В. Бєрлізов // Електротехнічні і комп`ютерні системи. – 2015. – № 19 (95). – С. 165 – 168.

Literatura

1. Durillo, J.J., Zhang, Y., Alba, E., Harman, M., Nebro, A.J. A study of the bi-objective next release problem // Empirical Software Engineering. – 2011. – Vol. 16 (1). – P. 29 – 60.
2. Harman, M., Jones, B.F. Search based software engineering // Information and Software Technology. – 2001. – № 43 (14). – P. 833 – 839.
3. Nogin, V. D. Decision making in the multicriteria environment: qualitative approach. – M. : FIZMATLIT, 2002. – 176 p.

4. Denisiuk, A. V., Liubchenko, V. V. Actual problems of software quality // Elektrotekhnicheslie i komputernye sistemy. – 2013. – № 9 (85). – P. 142 – 148.
5. Liubchenko, V. V., Berlizov E. V. The algorithm of finding Pareto-optimal solutions for next release problem // Elektrotekhnicheslie i komputernye sistemy. – 2015. – № 19 (95). – P. 165 – 168.

RESUME

V.V. Liubchenko

Decision support system for the next release problem

In the article, structural model of the decision support system for the next release problem is developed, and the models and techniques used to its implementation are defined.

The next release problem is formulated as the bi-objective optimization problem. It takes proper account of different values of different requirements for customers, different resources needs to implement different requirements, the existence of relationships between requirements.

Because of formulated bi-objective optimization problem is NP-hard, heuristic optimization techniques are applied to solve it. We describe the properties of solution sets obtained using different heuristic techniques and constraints of the using of heuristic techniques.

The structure of decision support systems is provided by the description of its components.

Taking into account the proposed models and techniques implementation of decision support systems provides for decision maker a wide range of different solutions to select the final one. It increases the number of potential analyzed solutions and reduces the influence of subjective factors on the selection of the final decision.

В.В. Любченко

Система підтримки прийняття рішень для задачі наступного релізу

У роботі розроблено структурну модель системи підтримки прийняття рішень для задачі наступного релізу та визначено моделі і техніки, які застосовуються для її реалізації.

Сформульовано задачу наступного релізу як задачу двокритеріальної оптимізації з урахуванням різної цінності вимог для різних споживачів, різних потреб ресурсів на реалізацію різних вимог, існування залежностей між вимогами.

Оскільки сформульована задача двокритеріальної оптимізації є NP-трудною, для її вирішення застосовуються евристики пошукової оптимізації. Описано властивості множин рішень, отриманих за допомогою різних евристичних технік, а також обмеження їх застосування.

Показано структуру системи підтримки прийняття рішень, надано опис її компонентів.

Реалізація системи підтримки прийняття рішень з урахуванням запропонованих моделей і технік забезпечує особі, що приймає рішення, широкий спектр припустимих рішень, з яких обирається остаточне рішення. При цьому збільшується кількість потенційних рішень, які аналізуються, а також знижується вплив суб’єктивного фактору на вибір остаточного рішення.

Надійшла до редакції 27.08.2015

 ©В.В. Любченко